AF IZW

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE

APPEAL BRIEF TRANSMITTAL	Docket Number: 10191/1333	Conf. No. 4178

Application Number	Filing Date	Examiner	Art Unit
09/527,424	March 17, 2000	Kimberly N. MCLEAN MAYO	2187

Invention Title	Inventor
METHOD AND DEVICE FOR SECURING DATA WHEN ALTERING THE STORAGE CONTENTS OF CONTROL UNITS	Rolf KOHLER et al.

Address to:

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:
Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on

Date: 2/16, 2005 Reg. No. 36,197

Signature: _____
Jong H. Lee

Further to the Notice of Appeal dated December 14, 2004 (filed at the PTO on December 16, 2004) for the above-referenced application, enclosed are three copies of an Appeal Brief. Accompanying the Appeal Brief is the Appendix to the Appeal Brief.

The Commissioner is hereby authorized to charge payment of the 37 C.F.R. § 1.17(c) appeal brief filing fee of **$500.00**, and any additional fees associated with this communication to the deposit account of **Kenyon & Kenyon**, deposit account number **11–0600**.

Dated: 2/16, 2005 By: _____ (by P. no. 36,197)

Richard L. Mayer (Reg. No. 22,490)
KENYON & KENYON
One Broadway
New York, N.Y. 10004
(212) 908-6479 (telephone)
(212) 908-6479 (telephone)

CUSTOMER NO. 26646
PATENT & TRADEMARK OFFICE

[10191/1333]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant(s)	:	Rolf KOHLER et al.
Application. No.	:	09/527,424
Filed	:	March 17, 2000
For	:	METHOD AND DEVICE FOR SECURING DATA WHEN ALTERING THE STORAGE CONTENTS OF CONTROL UNITS
Art Unit	:	2187
Examiner	:	Kimberly N. McLEAN MAYO
Conf. No.	:	4178

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on

Date: 2/16, 2005 Reg. No. 36,197

Signature: _____
Jong H. Lee

APPELLANTS' APPEAL BRIEF
UNDER 37 C.F.R. § 41.37

S I R :

Applicants filed a Notice of Appeal dated December 14, 2004 (filed at the PTO on December 16, 2004) appealing from the Final Office Action dated June 18, 2004, in which claims 1-38 of the above-identified

application were finally rejected. This Brief is submitted by Applicants in support of their appeal.

I. REAL PARTIES IN INTEREST

The above-identified Applicants and Robert Bosch GmbH of Stuttgart, Germany, are the real parties in interest.

II. RELATED APPEALS AND INTERFERENCES

No appeal or interference which will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal is known to exist to the undersigned attorney or is believed by the undersigned attorney to be known to exist to Applicants.

III. STATUS OF CLAIMS

Claims 1-38 are pending in this application. Among the claims on appeal, claims 1, 12, 24, 27 and 33-38 are independent. Claims 2-11 and 29-30 depend on claim 1; claims 13-23 and 31-32 depend on claim 12; claims 25-26 depend on claim 24; and claim 28 depends on claim 27.

IV. STATUS OF AMENDMENTS

No amendment has been made subsequent to the final Office Action mailed on June 18, 2004.

V. SUMMARY OF THE INVENTION

The present invention relates to a method and an associated device for programming data in a memory of a computer, wherein during programming, an identifier that identifies correct erasing and/or programming of the memory is entered into the memory. (Specification, p. 1, l. 2-5). The method and the respective device serve to secure the functionality of a control unit, for example, when an interruption or malfunction has occurred during erasing and/or programming. (P. 2, l. 22-

23). To do so, when an interruption or a reset occurs during erasing or programming or when the device otherwise becomes de-energized, this is noted in a memory device, in particular in the memory device to be programmed. (P. 2, l. 23-26). In addition, in programming, an identifier that identifies correct erasing and/or programming of the memory is entered into an area of the memory that is to be erased and/or programmed later, in particular an area that is to be erased and/or programmed last, and this identifier is altered before erasing or programming the data or programs in such a way that the program is not executed if programming is incomplete and/or the data is not used if data entry is not complete. (P. 2, l. 26 - p. 3, l. 1). Thus, an error in programming or erasing can be corrected after a possible data modification. (P. 3, l. 1-2).

Figure 1 shows as an example a hardware arrangement for implementing the method according to the present invention, which arrangement includes a computer system (control unit) 100, the control unit 100 in turn including a microprocessor 106 which is connected to additional elements by a bus system 109. (P. 4, l. 2-6). Various memory devices 104, 105 and 108 may be present in the control unit. (P. 4, l. 11-12). A volatile memory 105, e.g., in the form of RAM may be used, among other things, as buffer storage of data and/or programs. (P. 4, l. 12-13). Program routines may also be started from RAM 105. (P. 4, l. 13-14). Memory 104 is an erasable nonvolatile memory, e.g., a flash EPROM memory, and a plurality of flash EPROMs which differ greatly with regard to erasing and/or programming may be used here. (P. 4, l. 14-15). For example, a first type of flash EPROM 104 must be erased completely before reprogramming; a second type of flash EPROM 104 may be erased and/or programmed in blocks, for example; a third type of flash EPROM 104 may be programmed by pages, for example but no explicit erasing is necessary. (P. 4, l. 16-19). In addition to nonvolatile memory 104, another nonvolatile memory 108 may also be provided, e.g., it is possible to use a nonerasable read-only

memory ROM, and an erasable nonvolatile memory in the form of an EPROM, in particular another flash EPROM may also be used. (P. 4, l. 19-21). Only one programmable and/or erasable nonvolatile or refreshable memory, shown here in the form of element 104, is necessary to illustrate the present invention. (P. 4, l. 23-25). Block 107 represents possible peripheral components, i.e., interface elements for connecting external peripherals. (P. 4, l. 27-28). Additional elements and subassemblies represented, for example, by element 110, e.g., in the form of additional peripheral assemblies, memory elements, microprocessors or microcontrollers may optionally be connected to bus system 109. (P. 4, l. 28-30).

Although no means-plus-function claims are deemed to be presented in this Appeal (and even if they were presented, no such claims are argued separately in this appeal), to the extent the Board of Appeals may determine that claims 24, 27, 34, 37 and 38 recite means-plus-function limitations (e.g., "programming arrangement," "reprogramming arrangement," and "arrangement for reprogramming"), Applicants provide the following descriptions. Programming may be performed by insertion of a data carrier in control unit 100 itself or by using data from internal memories such as nonerasable nonvolatile memory 108. (Fig. 1; p. 4, l. 30 - p. 5, l. 2). In addition to this internal type of programming, however, an external programming unit 101, e.g., a second control unit or computer, may also be provided for programming and entering data either serially over interface 102 or as a parallel connection over a bus system 103. (P. 5, l. 2-5). With internal programming, a set of user programs and/or data may be loaded from a data memory described above such as memory 108 in the form of ROM into volatile memory 105 and/or into erasable nonvolatile memory 104 and optionally executed there. (P. 5, l. 5-8). Likewise, the programs and/or data may be entered with the help of data carriers into a peripheral element such as a reader for data carriers such as a diskette

drive or a CD-ROM drive that can be connected to a possible peripheral module 107. (P. 5, l. 8-10). The same possibilities are also available for external programming unit 101. (P. 5, l. 10-11).

Figure 2 shows in detail the erasable or programmable nonvolatile memory 104, which is divided into various areas 200 to 205, for example, and there is a cell or a memory section 206 in memory area 205. (P. 5, l. 13-15). One example allocation of the areas is as follows: areas 200-203 may contain vector tables and data, area 204 may contain a programming routine, and area 205 may contain at least one application program. (P. 5, l. 17-19). With internal programming, boot vectors with jumps to other memory areas of this memory or other memories may also be located in memory area 200, for example. (P. 5, l. 19-21). For example, certain switching functions are embedded in area 201, enabling input and output coordination of the memory with microprocessor 106, for example. (P. 5, l. 21-22). Area 202, for example, contains a start routine through which the remaining course is determined, depending on whether an application is to be run or programming is to take place. (P. 5, l. 22-24). Area 203 is optional and may contain additional data and/or routines, e.g., a program routine loaded by the startup program in area 202 into ROM 105, for example, where it controls the remaining programming operation, is stored in memory area 204. (P. 5, l. 24-27). The division into memory areas as shown in Figure 2 is optional because when using an external programming device 101, for example, some of areas 200-205 may be omitted. (P. 5, l. 27-29). Programming of nonvolatile memory 104 usually takes place from the lower memory area to the upper, area 206 being programmed last. (P. 5, l. 29-30).

Figure 3 shows a data and/or program section, in particular a program identifier 300 in memory 104. (P. 6, l. 1-2). Program identifier 300 is located, for example, in memory area 206 of program memory area 205,

because this is programmed late in the sequence of the programming operation, e.g., being programmed last. (P. 6, l. 2-4). This program identifier 300 may be subdivided into various subsections 301-304, which may include, for example, a predetermined number of bit positions or a column of a predetermined length of letters and numbers or any characters. (P. 6, l. 4-6). Before or during the programming operation, at least one subsection, e.g., subsection 301, is selected as identifier K and stored for later checking. (P. 6, l. 6-8). However, with data and/or programs that are already known or with repeating programming operations, a certain identifier may be specified in the form of a subsection. (P. 6, l. 8-10). In addition, however, identifier K may also be composed of various data sections and/or program sections or the subsections contained therein. (P. 6, l. 10-11). These subsections, which may then be combined to form identifier K, may be selected at very different positions in the data and/or programs. (P. 6, l. 11-13).

In an example embodiment, a data and/or program section that is easy to recover, such as program identifier 300, is selected as the basis for identifier K, so the subsection(s) functioning as the identifier can also be recovered easily. (P. 6, l. 15-17). Its precise position should also be known even after modification to permit checking of identifier K. (P. 6, l. 17-18).

Subsection 301, which was selected as identifier K, for example, may include any six characters, e.g., G47F03, and is selected on the basis of the fact that it is unambiguous, which means the precise position of this subsection in the data and/or programs need not be known precisely. (P. 6, l. 20-23). However, if the position of identifier K is known precisely as first subsection 301 of program identifier 300, and if the number of characters or bits is known precisely because it is predetermined, then it is possible to use an identifier K which occurs more frequently in the data and/or programs already programmed or to be programmed. (P. 6, l. 24-27).

Figure 4 shows a flow chart of how programming with safeguarding of a memory is performed as in the case of nonvolatile memory 104, for example, from a current program for operation of the control unit, e.g., from a driving program with a control unit for controlling operating sequences of a motor vehicle. (P. 7, l. 1-4). This means that when the control unit is turned on, the programming session is started, e.g., by software protocol, and the operating system is automatically changed and then programmed. (P. 7, l. 4-6).

The safeguard is provided by the fact that before erasing or programming the entire memory or a block or a page, depending on the type of flash EPROM used, an identifier at the end of the last existing block or the last page or the last cell of the entire memory, said identifier having been entered previously, which identifies correct erasing and/or programming of the memory, is rendered invalid by a change, e.g., by being erased or reprogrammed, in particular it is rendered unidentifiable, e.g., for a comparison query or a search query. (P. 7, l. 8-13). To do so, an existing part of the data and/or programs of the program identifier is used, for example, and therefore, this identifier K need not be programmed separately. (P. 7, l. 15-16). Only when the entire programming operation or the erase and reprogramming operation is concluded and identifier K has been entered again is the program valid again. (P. 7, l. 16-18). If there is any interruption, a reset, or if the device otherwise becomes de-energized during erasing and/or programming, the altered identifier K is not entered again, and thus the program is not valid and cannot be executed. (P. 7, l. 18-20). In addition, however, this interruption is noted in the flash EPROM itself that is used, because identifier K is invalid, leading to the result that in a restart it is necessary to first inquire whether the existing program is capable of running. (P. 7, l. 20-23). If identifier K is invalid and thus the program is incapable of running, the operating system may automatically be

called up for programming, in particular the boot block, in a nonvolatile nonerasable memory such as memory 108 in the form of a ROM or in a nonerasable part of flash EPROM memory 104 itself, and then the system waits for reprogramming. (P. 7, l. 23-26). Thus, an incompletely or incorrectly programmed flash EPROM 104 is not an obstacle that would prevent reprogramming. (P. 7, l. 26-27).

The operation starts in block 400 in Figure 4, and the program to be processed is determined first in block 401, which selection can be made from a number of different programs and program routines, by preselecting a program identifier 300 or by selecting a flag having one or more bit positions. (P. 7, l. 29 - p. 8, l. 3). This preselection is then checked in queries 402, 403 and 404. (P. 8, l. 3-4). Identifier K and/or the respective subsection(s) may also be specified in block 401, i.e., identifier K may thus be specified for all programming operations, e.g., preselected with repeating programming operations or per each programming operation and thus it may be variable on the whole. (P. 8, l. 7-10). In block 402 a query is then issued to determine whether driving program 1 has been selected by the program identification or the selection of flags. (P. 8, l. 10-12). If this is the case, the system returns to block 405, and a reset is triggered in block 405, performing the necessary initialization for driving program 1 and establishing the communication connection. (P. 8, l. 12-14). If it is found in query 402 that driving program 1 has not been selected, then a check is made in query 403, for example, to determine whether there is a programming request. (P. 8, l. 14-16). To do so, a query is issued at branching point 403 to determine whether the boot block or the boot program has been selected as a possible sign of subsequent programming. (P. 8, l. 16-17). If this is not the case, additional queries, with query 404 being shown as optionally representative, may check on which additional possible program has been selected. (P. 8, l. 17-19). If none of the programs available has been selected, one goes to the end of the process in

block 414. (P. 8, l. 19-20). Otherwise, it is possible to jump to additional programs starting from query 404 which is shown as representative for other branches. (P. 8, l. 20-22). However, if a boot program, e.g., boot program 1, is selected in query 403, this leads to block 410 where there is also a reset to the beginning (reset) and initialization depending on boot program 1 and establishment of communication. (P. 8, l. 24-26). In block 411, the session of the programming mode of the programming device is then started. (P. 8, l. 26-27). Next in block 412, identifier K is altered, in particular rendered unidentifiable as part of the program, e.g., as part of program identifier 300. (P. 8, l. 27-28). Before the change in identifier K, this and a possible flag for interruption of the programming operation may be analyzed. (P. 8, l. 29-30). Likewise, identifier K may be selected or predetermined here immediately before the change in identifier K by control unit 100 or a programming unit 101. (P. 8, l. 30 - p. 9, l. 1). For rapid erasing or programming, a change in baud rate can be implemented in block 412, switching from a lower normal communication bit rate to a higher erase and/or programming bit rate. (P. 9, l. 1-3). Then the actual erasing or programming operation takes place in block 413, erasing by pages, blocks, globally or by cells and/or reprogramming, depending on the memory used. (P. 9, l. 3-5). At the end of the programming operation in block 413 and/or after an interruption, e.g., due to a disturbance, a reset and/or a de-energized state of control unit 100 or microprocessor 106 of the programming operation, the correctness of identifier K is checked in block 415. (P. 9, l. 5-8). If identifier K is not recognized as correct, there is a renewed entry into the interrupted boot program. (P. 9, l. 8-9). It is possible in a block 416 to provide for a flag noting the cause and the time of the interruption, for example, to be entered into a memory, in particular the memory to be programmed. (P. 9, l. 9-11). However, if identifier K is identified as correct, this leads to block 414, the end of the process. (P. 9, l. 11-12). This means that even after complete programming without an interruption, identifier K and/or any flag that might be present can be

checked. (P. 9, l. 12-14). Since identifier K is entered into an area that is programmed as late as possible, preferably last, such as memory area 206 of programming area 205 in Figure 2, a correct identifier K is available again only when the programming operation has been finished completely and correctly. (P. 9, l. 14-17).

For security reasons, identifier K and optionally the flag for an interruption may also be checked and analyzed at the start of an application program, and optionally also at the beginning of a programming routine or a boot program, i.e., in queries 402, 403, 404. (P. 9, l. 19-22). Then it is possible to enter into an application program only if the correct program has been selected, and in addition identifier K and/or the flag for an interruption has been checked and recognized as correct. (P. 9, l. 22-24). If jumping to a boot program, e.g., a programming routine, i.e., if programming is to occur anyway, there is at least an indication if identifier K is not correct or there is a flag. (P. 9, l. 24-26). If the memory programming is to be secured and performed cell by cell, page by page or block by block, it is also possible to determine whether the identifier is correct and/or whether there is a flag with all the parts of the data and/or programs not to be programmed. (P. 9, l. 26-29).

Before the start of an erasing or programming operation, i.e., in block 412 in this example, this identifier K is altered, e.g., it is rendered unidentifiable. (P. 10, l. 1-2). The expected identifier K is advantageously part of the program itself. (P. 10, l. 2-3). If no boot program is selected in block 401 but instead a driving program, for example, is selected, i.e., an application program, this is recognized in queries 402, 403, 404 and optionally following queries. (P. 10, l. 3-5). In the case of driving program 1, query 402 then leads to block 405 where the above-mentioned operations are carried out. (P. 10, l. 5-6). Then following the functions of processing in the application program, for example, query 406 determines whether or not a programming session is to

be started. (P. 10, l. 6-8). If this is not the case, it goes to block 408 where the selected program which is already in a memory, in particular nonvolatile memory 104, is processed, arriving at the end at block 414 again, the end of the process. (P. 10, l. 8-10). However, if query 406 reveals that the operator wishes to perform programming within the application program, in particular driving program 1, this leads to block 407. (P. 10, l. 10-12). Depending on certain conditions, the programming mode session is then started in block 407, which conditions include, for example, the engine not running, e.g., in conjunction with a default mode, consent that the syntax is in order, in particular when a baud rate switch is requested, etc. (P. 10, l. 12-15). These conditions can also be queried in block 411. (P. 10, l. 15). Depending on maintenance of the conditions, different error messages are issued in detail. (P. 10, l. 15-16). If these conditions for the programming mode are met, the source and target address are saved in another memory or memory area and a software reset is initiated. (P. 10, l. 16-18). At this point a code may also be set, indicating that programming of the nonvolatile memory, in particular flash memory 104, is to be performed. (P. 10, l. 18-19). If a software reset occurs then, initialization is performed after the reset and the code entered for flash programming is checked. (P. 10, l. 19-21). If this code is correct, it may also be altered in a controlled manner, for example, in order to be identified again later on the basis of this action. (P. 10, l. 21-22). Next, there is a function call for establishing communication in block 409. (P. 10, l. 26). To do so, this must not already be present and the code that has been altered in a controlled manner must be recognized. (P. 10, l. 27-28). Then initialization takes place in block 409 as it does at the start of communication in the boot program or in the boot block in block 410. (P. 10, l. 28-30). The interface is likewise initialized at a normal communication transfer rate, the transmission is set, communication is activated, etc. (P. 10, l. 30 - p. 11, l. 1). Likewise, when communication is established, a flag, for example, is set for the baud rate switch, so that immediately after sending a positive response in the course of communication, the transmission rate or baud rate in block 412 can be switched again. (P. 11, l. 1-4). Likewise, the programming mode session is activated here. (P. 11, l. 4). After a positive response in the wake of the communication connection, this is

entered into a buffer and the transmission process begins, after which the system goes to block 412 and the remaining course in the boot program begins again with a baud rate change. (P. 11, l. 4-7).

VI. GROUNDS OF REJECTION TO BE REVIEWED

The following grounds of rejection are presented for review on appeal in this case:

(A) Whether claims 1, 3-12, 14-24, 26-27 and 29-38 are anticipated under 35 U.S.C. § 102(e) by U.S. Patent No. 6,000,004 to Fukumoto ("Fukumoto").

(B) Whether claims 2, 13, 25 and 28 are rendered obvious under 35 U.S.C. § 103(a) by U.S. Patent 6,000,004 ("Fukumoto").

VII. GROUPING OF CLAIMS

For each ground of rejection, all claims subject to the rejection do not stand or fall together. For the anticipation rejection of claims 1, 3-12, 14-24, 26-27 and 29-38 based on Fukumoto, claims 1, 3-12, 14-24, 26-27 and 29-32 will be argued as one group, and claims 33-38 will be argued as another group. For the obviousness rejection of claims 2, 13, 25 and 28 based on Fukumoto, all claims will be argued as a single group.

VIII. ARGUMENTS

A. Claims 1, 3-12, 14-24, 26-27 and 29-38

Claims 1, 3-12, 14-24, 26-27 and 29-38 stand rejected under 35 U.S.C. §102(e) as being anticipated by United States Patent No. 6,000,004 to Fukumoto (the "Fukumoto reference"). For at least the following reasons, Applicants respectfully submit that the anticipation rejection of Claims 1, 3-12, 14-24, 26-27 and 29-38 based on the Fukumoto reference should be reversed.

i. Claims 1, 3-12, 14-24, 26-27 and 29-32

Claim 1 recites:

A method of programming information in a memory arrangement of a computer, comprising the steps of:

providing an identifier into an area of the memory arrangement that is to be programmed, the identifier identifying a correct programming of the memory arrangement; and

altering the identifier in the memory arrangement before programming the information.

In support of the rejection, the Examiner contends in the Final Office Action that the "block protect indicator taught by Fukumoto indicates **a correct programming** and erasing ***area*** by indicating that **programming** and erasing **is allowed** to the corresponding region (col. 11, l. 7-11)." (Final Office Action, p. 5, "Response to Arguments" section). Applicants note that the Examiner is once again reverting to the claim interpretation previously raised in the Advisory Action mailed on March 5, 2003, i.e., the Examiner once again contends that the phrase "correct programming of the memory arrangement" means "an approved area to program." However, the Examiner's interpretation is incorrect for at least two reasons. First, in contrast to the Examiner's interpretation, claim 1 does not recite that the identifier identifies a correct programming ***area***; rather, claim 1 clearly recites that the identifier identifies "a **correct programming** of the memory arrangement." To the extent the Examiner is ignoring the explicit words of the claim, the Examiner's interpretation is in violation of the fundamental rule that an anticipatory reference must ***identically disclose each and every claim element***. See Lindeman Machinenfabrik v. American Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984). Second, the Examiner's interpretation clearly contradicts the meaning of the phrase "a **correct programming** of the memory arrangement" as described in the specification. The second issue will be explained in further detail below.

To anticipate a claim under § 102(b), a single prior art reference must identically disclose each and every claim element. See Lindeman

Machinenfabrik v. American Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984). If any claimed element is absent from a prior art reference, it cannot anticipate the claim. See Rowe v. Dror, 112 F.3d 473, 478 (Fed. Cir. 1997). Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claim invention, arranged exactly as in the claim. Lindeman, 703 F.2d 1458 (Emphasis added). Additionally, not only must each of the claim limitations be identically disclosed, an anticipatory reference must also enable a person having ordinary skill in the art to practice the claimed invention, namely the inventions of the rejected claims, as discussed above. See Akzo, N.V. v. U.S.I.T.C., 1 U.S.P.Q.2d 1241, 1245 (Fed. Cir. 1986). To the extent that the Examiner may be relying on the doctrine of inherent disclosure for the anticipation rejection, the Examiner must provide a "basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristics necessarily flows from the teachings of the applied art." (See M.P.E.P. § 2112; emphasis in original; see also Ex parte Levy, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990)).

With respect to the issue of claim interpretation, Applicants note that even if one applied the Federal Circuit cases most favorable to the Examiner's position, even if the ordinary and customary meaning of a claim term to persons skilled in the pertinent art is evident, "the inventor's written description of the invention, for example, is relevant and controlling insofar as it provides clear lexicography or disavowal of the ordinary meaning." C. R. Bard Inc. v. United States Surgical Corp., 73 U.S.P.Q.2d 1011, 1014 (Fed. Cir. 2004). This rule is clearly consistent with the long-standing rules that "Applicant may be his or her own lexicographer as long as the meaning assigned to the term is not repugnant to the term's well known usage," and that the broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. MPEP 2111. Furthermore, the pending claims cannot be interpreted by reading limitations of the specification into a claim, to thereby narrow the scope of the claim. MPEP 2111.

Initially, Applicants respectfully submit that there is absolutely no support for the Examiner's contention that the ***ordinary and customary meaning*** of the phrase "correct programming of the memory arrangement" is "an approved area to program." If anything, the ordinary and customary meaning of the phrase "correct programming of the memory arrangement" is that the "programming of the memory arrangement" was performed without fault or error: "correct usually implies freedom from fault or error." (Merriam Webster's Collegiate Dictionary, 10th Ed. (1993) (defining "correct" as an adjective).

Furthermore, even if the Examiner could somehow show that the ordinary and customary meaning of the phrase "correct programming of the memory arrangement" is "an approved area to program" (for which proposition the Examiner has not provided any objective support), Applicants' own definition of "a correct programming of the memory arrangement" is "***controlling insofar as it provides clear lexicography or disavowal of the ordinary meaning.***" C. R. Bard Inc. v. United States Surgical Corp., 73 U.S.P.Q.2d 1011, 1014 (Fed. Cir. 2004). The meaning of the phrase "a correct programming of the memory arrangement" is clearly described throughout Applicants' Specification to be consistent with the interpretation asserted by Applicants, i.e., ***programming operation has been completed correctly in the memory arrangement***. For example, the Specification indicates the following:

> The method and the respective device serve to secure the functionality of a control unit, for example, when an interruption or malfunction has occurred during erasing and/or programming. To do so, when an interruption or a reset occurs during erasing or programming or when the device otherwise becomes de-energized, this is noted in a memory device, in particular in the memory device to be programmed. **In addition, in programming an identifier that identifies correct erasing and/or programming of the**

> **memory is entered into an area of the memory that is to be erased and/or programmed later**, in particular an area that is to be erased and/or programmed last, **and this identifier is altered before erasing or programming the data or programs in such a way that the program is not executed if programming is incomplete and/or the data is not used if data entry is not complete.** Thus, an error in programming or erasing can be corrected after a possible data modification.
>
> Advantageously, **the expected identifier that identifies the completeness and accuracy of the programming** is used as part of the program, in particular as part of the program identifier itself, and consequently does not take up any additional memory. (Specification, p. 3, l. 23 - p. 4, l. 6, *emphasis added*).

Thus, when viewed in light of the Specification, it is clear that the phrase "correct programming" as used in the claims of the present application means that a ***programming operation has been completed correctly***, not "an approved area to program" as asserted by the Examiner.

In view of the above explanation, it is quite clear that the block protect (BP) storage region and the erase complete (EC) storage region described in the Fukumoto reference do not teach or suggest "providing an **identifier** into an area of the memory arrangement that is to be programmed, the **identifier identifying a correct programming** of the memory arrangement," as recited in claim 1. The BP storage region is provided "in order to protect the data stored in each block. In this case, if the BP data is stored in the BP data storage region, then the erase and the write of the data from/into the block are prohibited in principle." Fukumoto, col. 4, ll. 40-43. Furthermore, the section of Fukumoto cited by the Examiner, col. 11, l. 7-11, merely indicates that "the **BP data storage region 1a disables the erase and the write of data** from /into the block 1 (to which the region 1a belongs) and thereby protects the data

stored therein, **if the BP (block protect) data has been stored in the BP data storage region 1a**." Thus, the BP data does not serve to identify a correct programming of the memory arrangement as claimed and described in the Applicants' specification, e.g., in the above-quoted section of p. 3, l. 23 - p. 4, l. 6. In addition, the data in the EC storage area of Fukumoto is erased during an erase operation. Fukumoto, col. 14, ll. 9-11. After the erase operation is complete, the EC data is written to the EC storage area. Fukumoto, col. 14, ll. 11-13. The EC data is only rewritten during an erase operation. Fukumoto, col. 13-15. Thus, the EC storage data does not serve to identify a correct programming of the memory arrangement.

Independent claims 12, 24, and 27 recite features similar to the above-discussed feature of claim 1 regarding an identifier identifying a correct programming of the memory arrangement. For at least the reasons stated above, the Fukumoto reference does not disclose each and every feature recited in independent claims 1, 12, 24, and 27, as well as dependent claims 3-11, 14-23, 26, and 29-32. It is therefore respectfully requested that this rejection be reversed.

ii. Claims 33-38

Claim 33 recites:

> A method of erasing information in a memory arrangement of a computer, comprising:
>
> providing an identifier into an area of the memory arrangement that is to be erased, the identifier identifying a correct erasing of the memory arrangement; and
>
> altering the identifier in the memory arrangement before erasing the information.

The Fukumoto reference does not disclose "altering the identifier in the memory arrangement before erasing the information." According to the Fukumoto reference, the EC storage area is erased during the erase operation,

and the data stored in the EC storage area is rewritten only when the erase operation is performed. Fukumoto, col. 14, ll. 9-15. However, the Examiner contends that the claimed feature of "**altering the identifier in the memory arrangement before erasing the information**" is met by Fukumoto since "the identifier is altered **via the altering of BP during a previous erase or program operation before a current erasing or programming of information**." (Final Office Action, p. 4). In this regard, the Examiner is contending that the claimed features of claim 33 are met by Fukumoto since the BP storage area contains an "identifier identifying a correct erasing of the memory arrangement," and rewriting of the BP storage region in the previous erase or program operation satisfies the feature of "altering the identifier in the memory arrangement before erasing the information." Applicants note that the Examiner's interpretation is completely incorrect, for the following reasons.

Initially, Applicants note that the BP data does not serve to identify a correct programming of the memory arrangement as claimed and described in the Applicants' specification. As noted previously, the BP data storage region 1a merely "**disables the erase and the write of data** from/into the block 1 (to which the region 1a belongs) and thereby protects the data stored therein, if the BP (block protect) data has been stored in the BP data storage region 1a." (Col. 11, l. 7-11). Furthermore, to the extent the Examiner is contending that rewriting of the BP storage region in the previous erase or program operation satisfies the claimed feature of "altering the identifier in the memory arrangement before erasing the information," the Examiner has misinterpreted the claimed features of claim 33: the Examiner's interpretation completely ignores the fact that the "**erasing**" recited in the feature of "the identifier identifying **a correct erasing** of the memory arrangement" and the "**erasing**" recited in the feature of "altering the identifier in the memory arrangement **before erasing** the information" refer to **the same "erasing" step**. For at least these reasons, Fukumoto does not teach the claimed invention of claim 33.

In addition, and independent of the above, the Examiner's contention that the "**altering of BP during a previous erase or program**

operation" as taught by Fukumoto satisfies the claimed feature of **"altering the identifier in the memory arrangement before erasing the information**" completely contradicts the Examiner's concurrent assertion that **BP** of Fukumoto **indicates** "that programming and erasing is allowed to the corresponding region": since Fukumoto indicates that "the **protected state is released by rewriting the BP data** stored in the BP data storage region" (col. 14, l. 5-7), if indeed the BP data are rewritten **during a previous erase and/or programming operation** as alleged by the Examiner, then the BP data storage region 1a **does not** provide the block protect function of disabling "the erase and the write of data from/into the block 1" (col. 11, l. 7-11) **for the current erase and/or programming operation**.

For the foregoing reasons, neither the BP storage data operation nor the EC storage data operation of Fukumoto teaches or suggests the claimed features of providing an "identifier" that identifies "a correct **erasing** of the memory arrangement," and "altering the identifier in the memory arrangement before **erasing** the information," as recited in claim 33. Since the Fukumoto reference does not disclose at least these features, the Fukumoto reference does not anticipate Claim 33. It is respectfully submitted that the anticipation rejection of claim 33 should be reversed.

Claim 34 recites a feature similar to the above-recited feature of claim 33 regarding altering the identifier before erasing the information. For at least the reasons described above in connection with claim 33, the Fukumoto reference does not anticipate new claim 34. It is respectfully submitted that the anticipation rejection of claim 34 should be reversed.

Claim 35 recites:

A method of erasing and programming information in a memory arrangement of a computer, comprising:

providing an identifier into an area of the memory arrangement that is to be erased and programmed, the

identifier identifying a correct erasing and programming of the memory arrangement; and

altering the identifier in the memory arrangement before erasing and programming the information.

As discussed above in connection with claim 33, the Fukumoto reference does not disclose an identifier identifying a correct programming of the memory arrangement, let alone disclose an "identifier identifying a correct erasing and programming of the memory arrangement" or "altering the identifier . . . before erasing and programming the information." For these reasons, the Fukumoto reference does not anticipate claim 35. It is respectfully submitted that the anticipation rejection of claim 35 should be reversed.

Claims 36-38 recite features similar to the feature of claim 35 regarding an identifier identifying a correct erasing and programming of the memory arrangement. For at least the reasons described above in connection with claims 33 and 35, the Fukumoto reference does not anticipate claims 36-38. It is respectfully submitted that the anticipation rejection of claims 36-38 should be reversed.

B. Claims 2, 13, 25 and 28

Claims 2, 13, 25 and 28 stand rejected under 35 U.S.C. §103(a) as being unpatentable over the Fukumoto reference. For at least the following reasons, Applicants respectfully submit that the rejection of claims 2, 13, 25 and 28 based on the Fukumoto reference should be reversed.

Claims 2, 13, 25, and 28 depend from independent claims 1, 12, 24, and 27, respectively. Independent claim 1 recites:

A method of programming information in a memory arrangement of a computer, comprising the steps of:

providing an identifier into an area of the memory arrangement that is to be programmed, the identifier identifying a correct programming of the memory arrangement; and

altering the identifier in the memory arrangement before programming the information.

Independent claims 12, 24, and 27 recite features similar to the above-recited features of claim 1 regarding an identifier identifying a correct programming of the memory arrangement.

In support of the rejection of independent claim 1, the Examiner contended that the "block protect indicator taught by Fukumoto indicates **a correct programming** and erasing ***area*** by indicating that **programming** and erasing **is allowed** to the corresponding region (col. 11, l. 7-11)." (Final Office Action, p. 5, "Response to Arguments" section). Applicants note that the Examiner is contending that the phrase "correct programming of the memory arrangement" means "an approved area to program." However, the Examiner's interpretation is not only unsupported by any extrinsic evidence, but clearly contradicts the meaning of the phrase "a **correct programming** of the memory arrangement" as described in the specification, for the reasons explained in detail below.

In order for a claim to be rejected for obviousness under 35 U.S.C. § 103(a), not only must the prior art teach or suggest each element of the claim, but the prior art must also suggest combining the elements in the manner contemplated by the claim. See Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 934 (Fed. Cir. 1990), cert. denied, 111 S. Ct. 296 (1990); In re Bond, 910 F.2d 831, 834 (Fed. Cir. 1990). "The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination." MPEP 2143.01 (citing In re Mills, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990)).

With respect to the issue of claim interpretation, Applicants note that even if one applied the Federal Circuit cases most favorable to the Examiner's position, even if the ordinary and customary meaning of a claim term to persons skilled in the pertinent art is evident, "the inventor's written description of the invention, for example, is relevant and controlling insofar as it provides clear lexicography or disavowal of the ordinary meaning." C. R. Bard Inc. v. United States Surgical Corp., 73 U.S.P.Q.2d 1011, 1014 (Fed. Cir. 2004).

Applicants respectfully submit that there is absolutely no support for the Examiner's contention that the ***ordinary and customary meaning*** of the phrase "correct programming of the memory arrangement" is "an approved area to program." Even if the Examiner could somehow show that the ordinary and customary meaning of the phrase "correct programming of the memory arrangement" is "an approved area to program" (for which proposition the Examiner has not provided any objective support), Applicants' own definition of "a correct programming of the memory arrangement" is "***controlling insofar as it provides clear lexicography or disavowal of the ordinary meaning.***" C. R. Bard Inc. v. United States Surgical Corp., 73 U.S.P.Q.2d 1011, 1014 (Fed. Cir. 2004). The meaning of the phrase "a correct programming of the memory arrangement" is clearly described throughout Applicants' Specification to be consistent with the interpretation asserted by Applicants, i.e., ***programming operation has been completed correctly in the memory arrangement***, not "an approved area to program" as asserted by the Examiner.

The block protect (BP) storage region and the erase complete (EC) storage region described in the Fukumoto reference do not teach or suggest "providing an **identifier** into an area of the memory arrangement that is to be programmed, the **identifier identifying a correct programming** of the memory arrangement," as recited in claim 1. The BP storage region is provided "in order to protect the data stored in each block. In this case, if the BP data is stored in the BP data storage region, then the erase and the write of the data from/into the block are prohibited in principle." Fukumoto, col. 4, ll. 40-43. Furthermore, the section of Fukumoto cited by the Examiner, col. 11, l. 7-11,

merely indicates that "the **BP data storage region 1a disables the erase and the write of data** from /into the block 1 (to which the region 1a belongs) and thereby protects the data stored therein, **if the BP (block protect) data has been stored in the BP data storage region 1a**." Thus, the BP data does not serve to identify a correct programming of the memory arrangement as claimed and described in the Applicants' specification, e.g., in the above-quoted section of p. 3, l. 23 - p. 4, l. 6. In addition, the data in the EC storage area of Fukumoto is erased during an erase operation. Fukumoto, col. 14, ll. 9-11. After the erase operation is complete, the EC data is written to the EC storage area. Fukumoto, col. 14, ll. 11-13. The EC data is only rewritten during an erase operation. Fukumoto, col. 13-15. Thus, the EC storage data does not serve to identify a correct programming of the memory arrangement.

Independent claims 12, 24, and 27 recite features similar to the above-discussed feature of claim 1 regarding an identifier identifying a correct programming of the memory arrangement. For at least the reasons stated above, the Fukumoto reference does not disclose each and every feature recited in independent claims 1, 12, 24, and 27. Since the Fukumoto reference does not anticipate claims 1, 12, 24 and 27 as described above, the Fukumoto reference cannot render dependent claims 2, 13, 25, and 28 obvious under 35 U.S.C. §103(a). It is therefore respectfully requested that this rejection be reversed.

IX. CONCLUSION

For the foregoing reasons, it is respectfully submitted that the final rejection of claims 1-38 should be reversed.

Respectfully submitted,

KENYON & KENYON

Dated: 2/16, 2005

By: _____

Richard L. Mayer
Reg. No. 22,490
CUSTOMER NO. 26646
PATENT & TRADEMARK OFFICE

R. no. 36,197